
Combinational Circuit Review

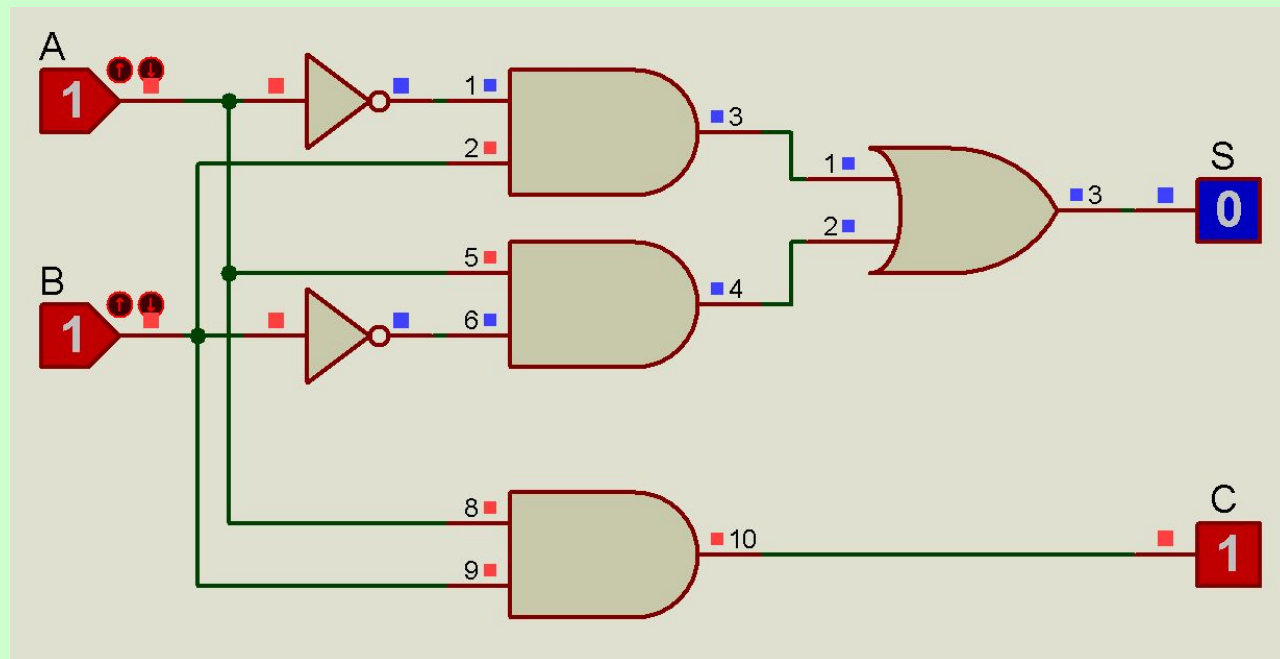
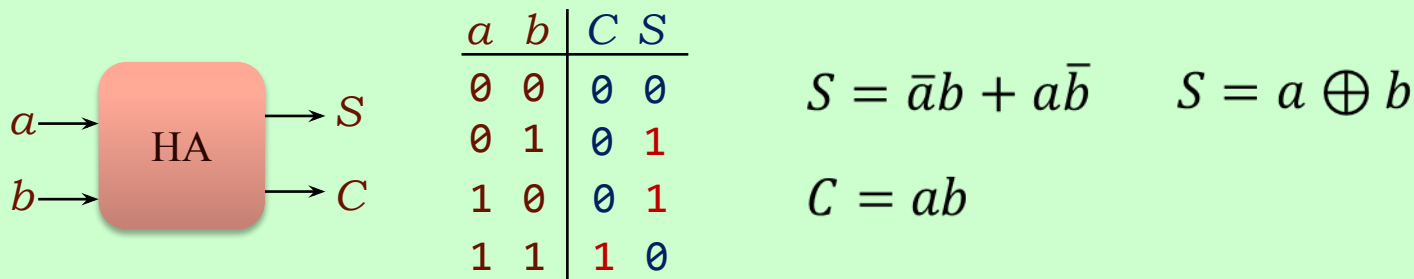
Combinational Circuit Design

Steps of designing Combinational circuits

- 1.** Determine how many inputs and outputs are required
- 2.** Describe the circuit function with Truth Table
- 3.** Derive logical equation for each output using
 - 1.** Sum-Of-Products (SOP) or
 - 2.** Product-Of-Sum (POS)
- 4.** Minimize the expression using one of the following methods
 - Algebraic rules
 - Karnaugh Map
 - Quine-McCluskey (which can be in the form of computer program)
- 6.** Implement (convert equations into equivalent logic gates)

Combinational Circuit Design Example

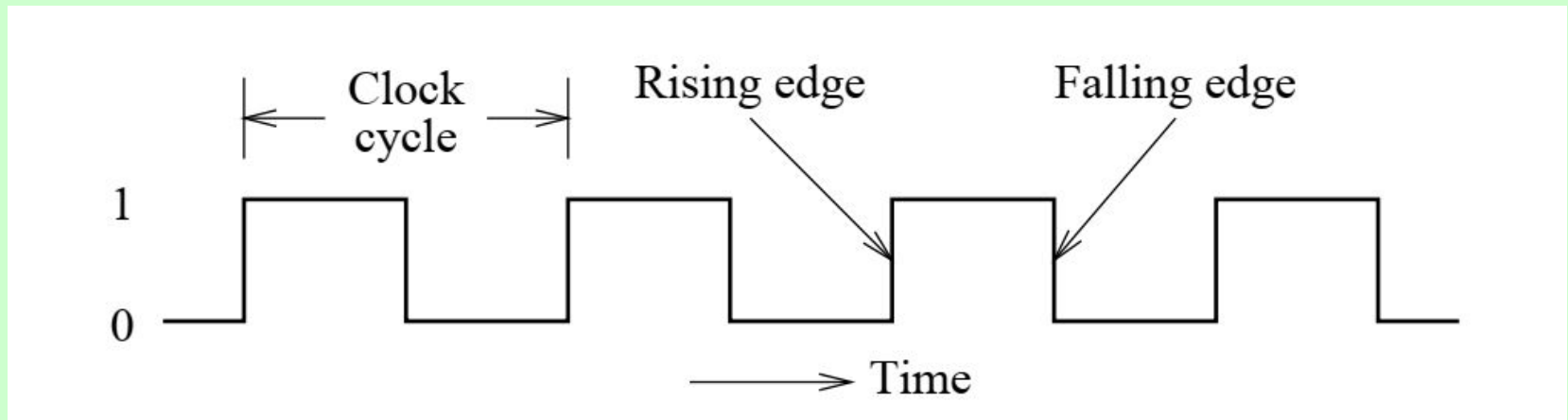
Design a half-adder (HA) circuit with two inputs (a,b) and two outputs sum (S) and carry (C). The HA function is to perform binary addition to its input



Synchronous & Asynchronous Systems

- Digital circuits can operate in *asynchronous* or *synchronous* mode
- *Asynchronous* - The time at which a change occurs in one circuit has no relation to the time a change occurs in another circuit (**not synchronized**)
- *Synchronous* - In this mode, all circuits in the system change their state at some precisely defined instants
- **The clock** signal in a digital system provides such a global definition of time instants at which changes can take place.
- The clock signal also specifies the speed at which a circuit can operate.

Clock Signal



- A clock is a sequence of 1s and 0s
- We refer to the period during which the clock is 1 as the ON period and the period with 0 as the OFF period
- The clock signal edge going from 0 to 1 is referred to as the rising edge (positive edge)
- The clock signal edge going from 1 to 0 is referred to as the falling edge (negative edge)

Clock Signal

- A clock cycle/period “ T ” is defined as the time between two successive rising (or falling) edges
- Clock rate or frequency “ f ” is measured in number of cycles per second (Hz)

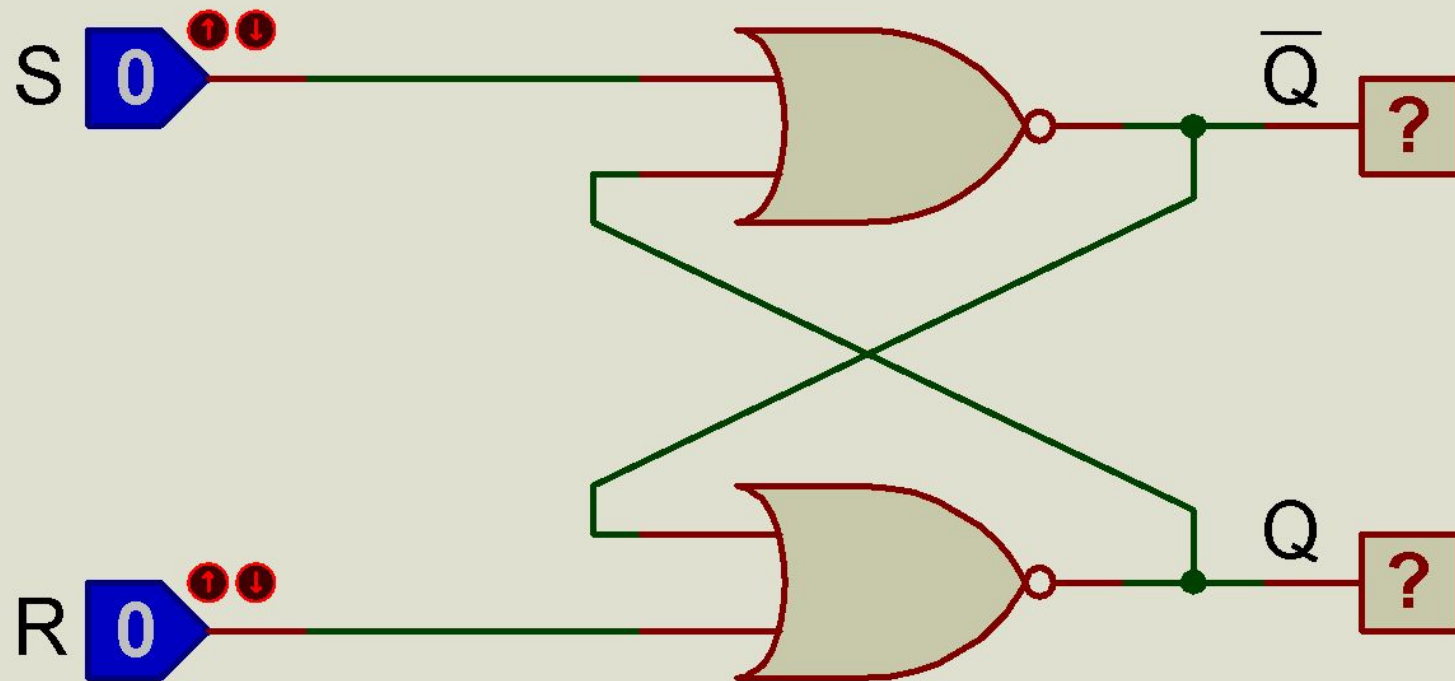
$$f = \frac{1}{T}$$

- The clock signal serves two distinct purposes in a digital circuit where it provides
 - The global synchronization signal for the entire system
 - Timing information in the form of a clock period

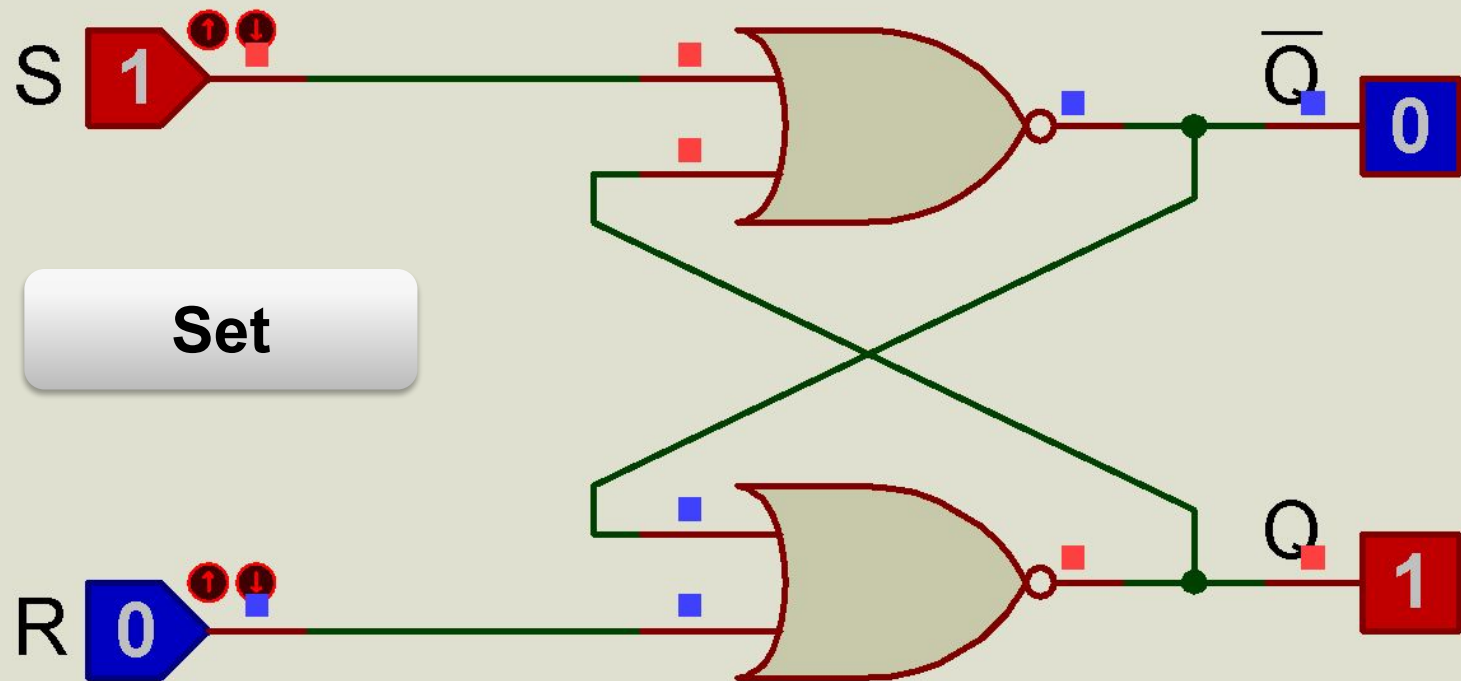
Sequential Elements (Latch & Flip Flop)

- Latches/Flip-Flops are the basic elements of sequential systems
- Latch or flip-flop represents 1-bit memory
- Latches are level-sensitive devices in that the device responds to the input signal levels (high or low).
- In contrast, flip-flops are edge-triggered. That is, output changes only at either the rising or falling edge

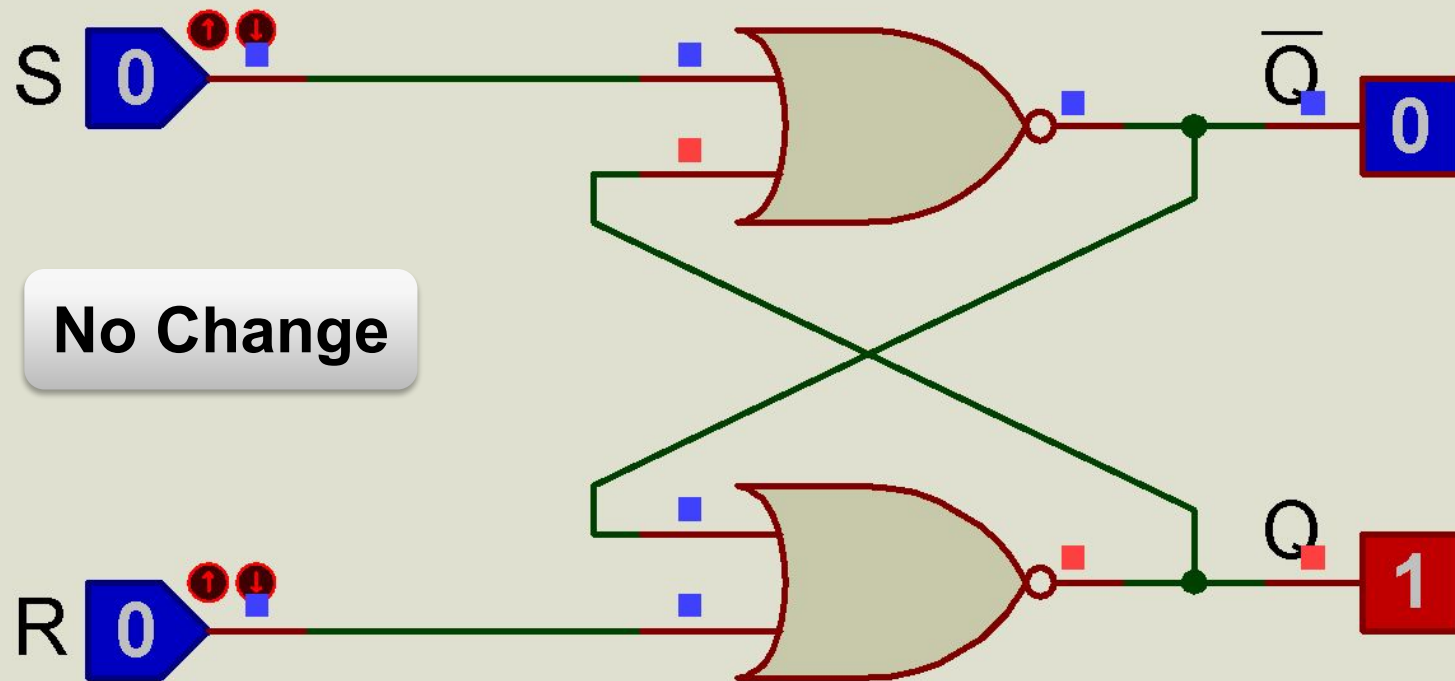
S-R Latch using NOR Gates



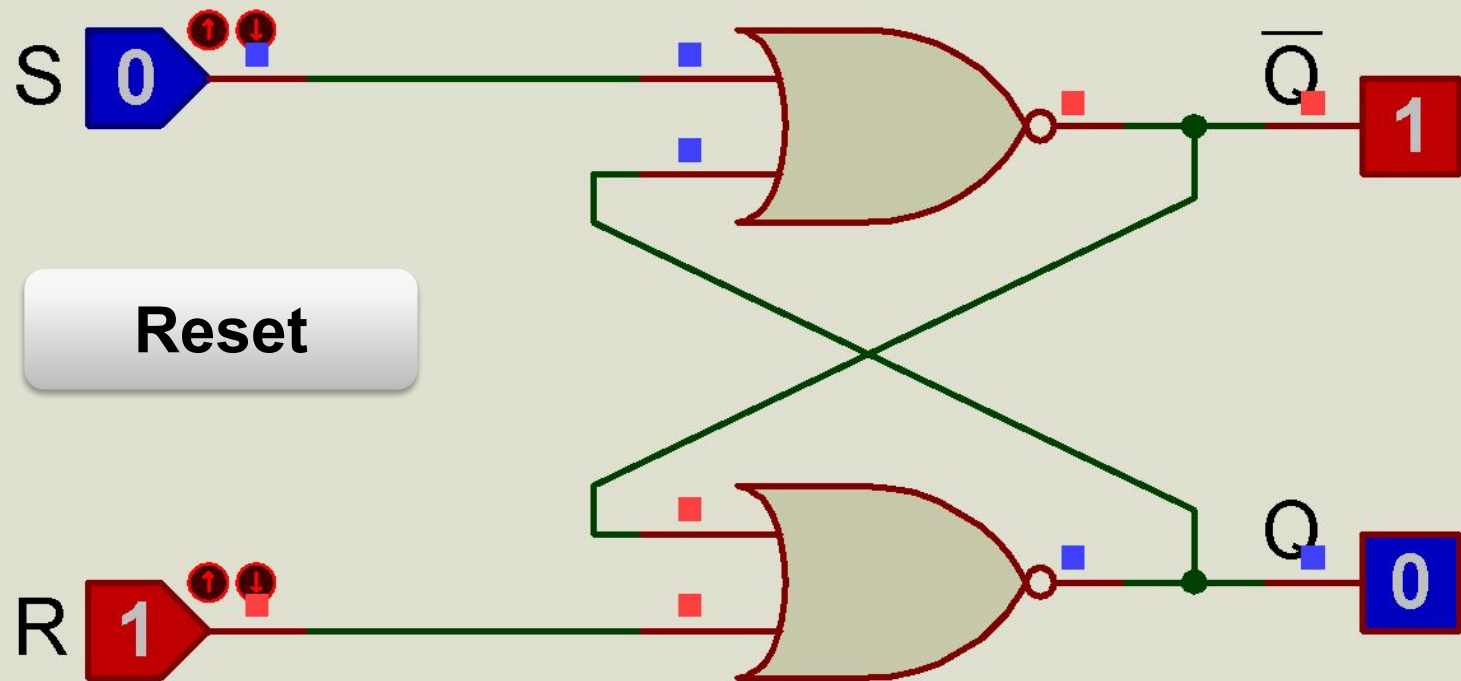
S-R Latch using NOR Gates



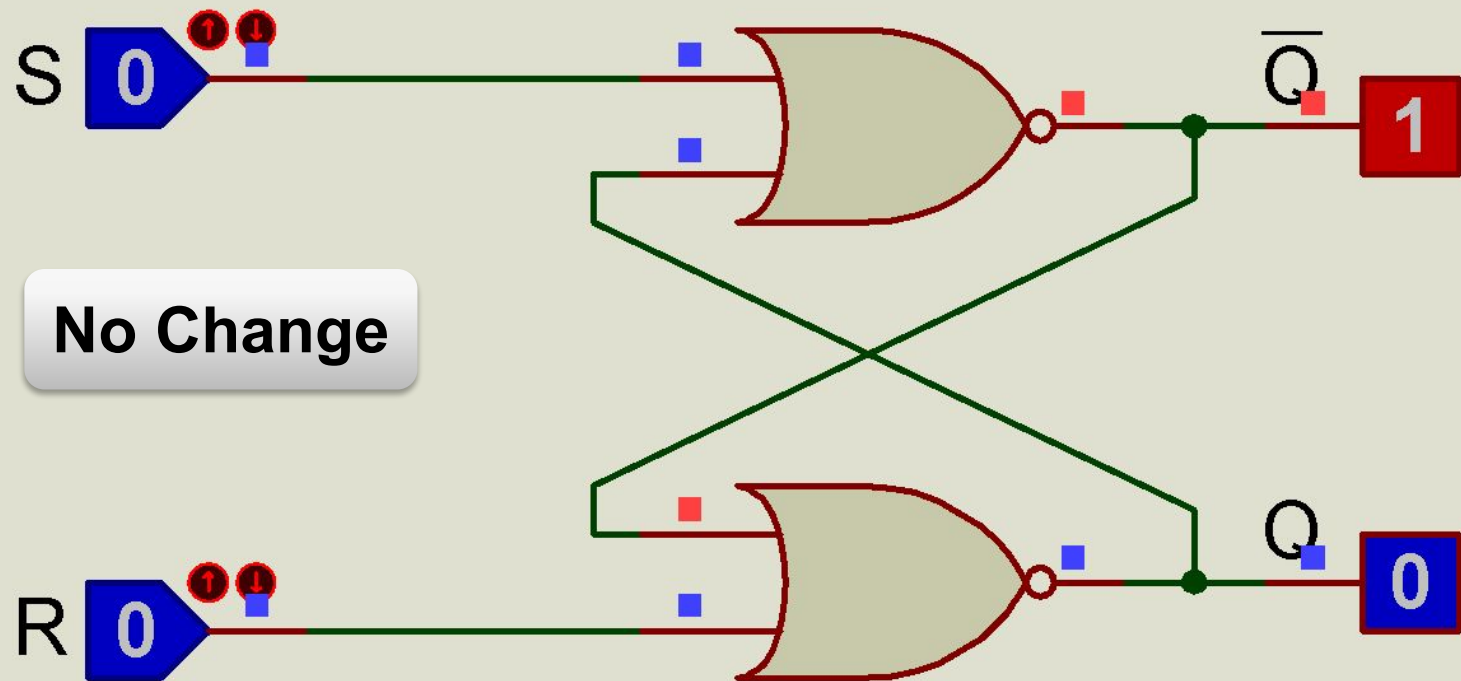
S-R Latch using NOR Gates



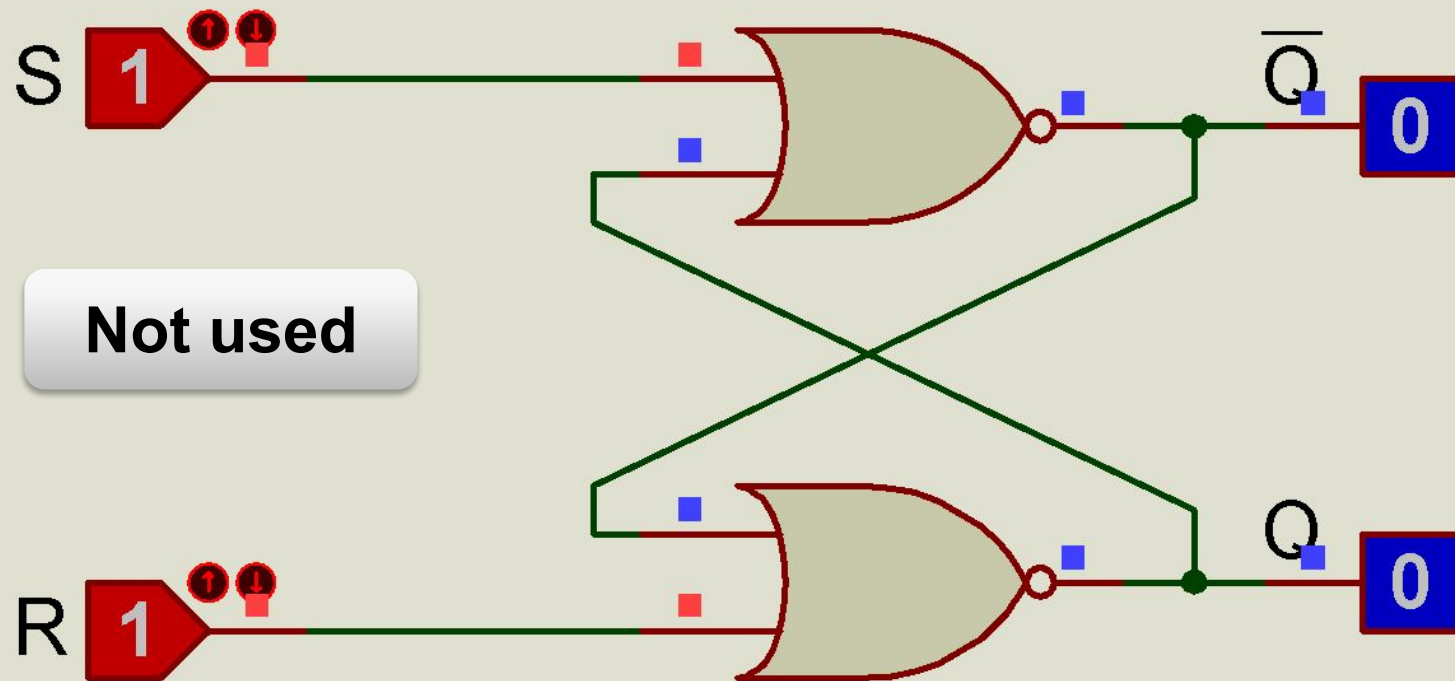
S-R Latch using NOR Gates



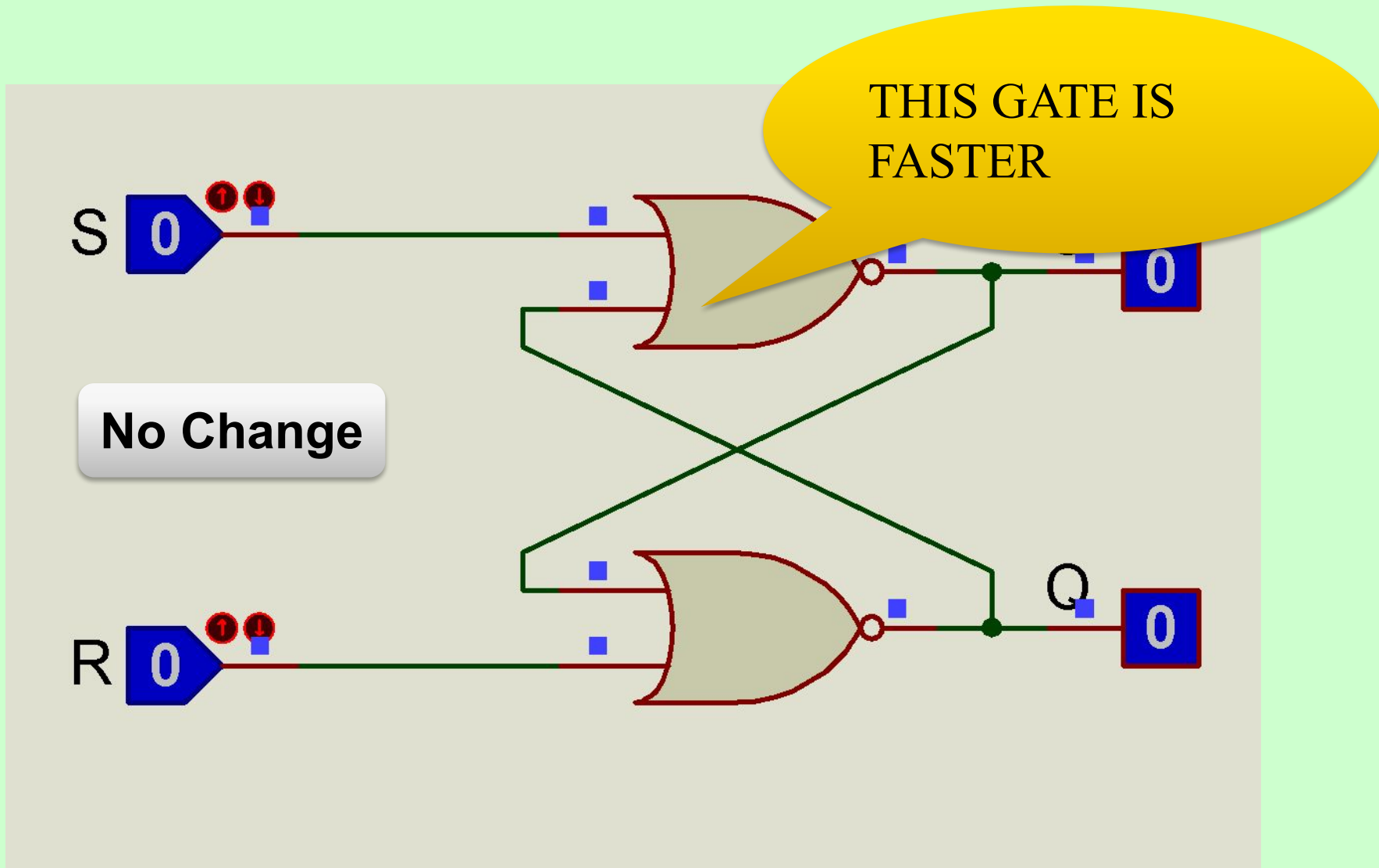
S-R Latch using NOR Gates



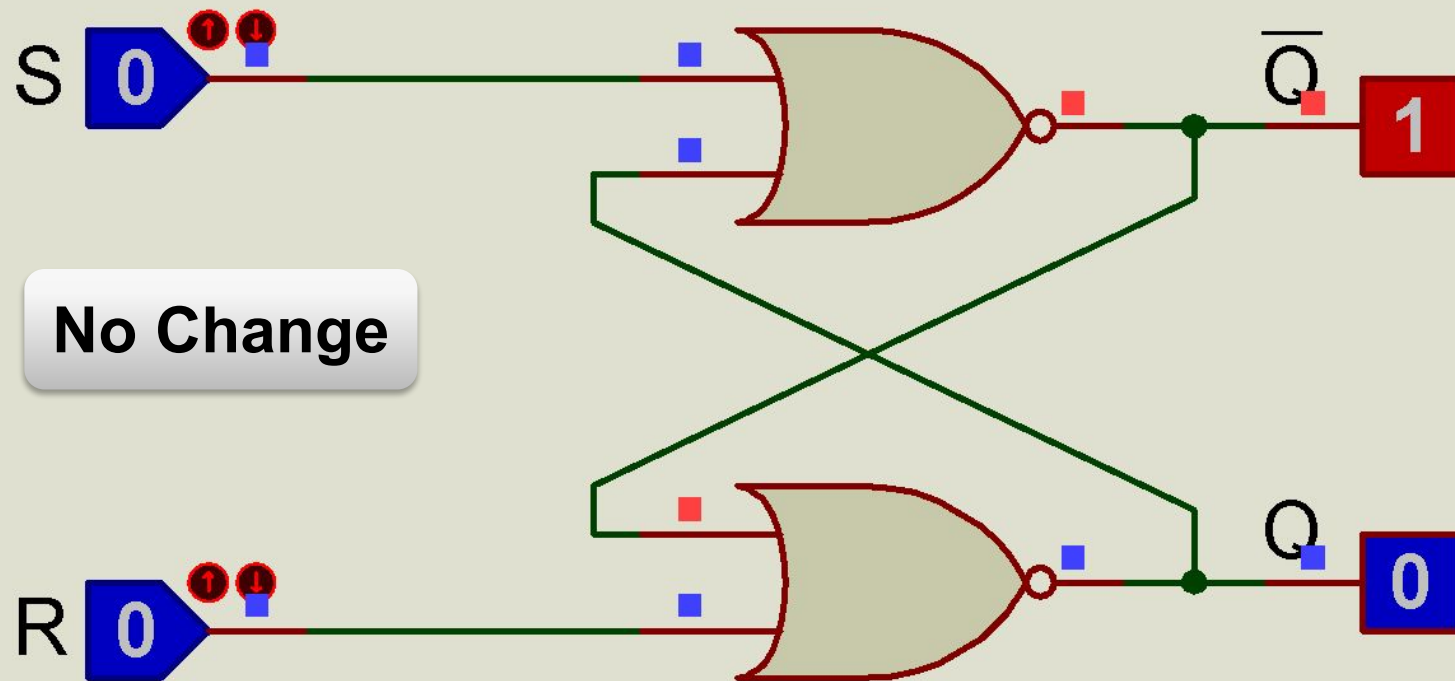
S-R Latch using NOR Gates



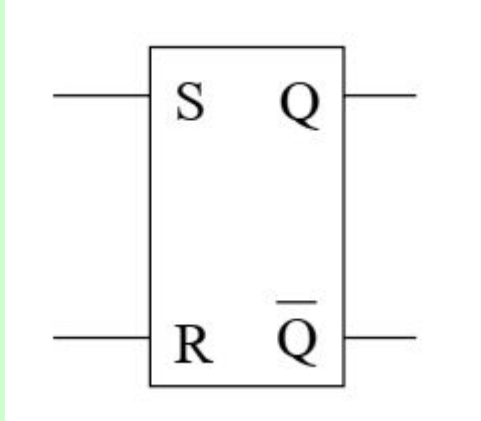
S-R Latch using NOR Gates



S-R Latch using NOR Gates



S-R Latch Symbol and Truth Table



S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	0

No
Change
Reset

Set

Hazard

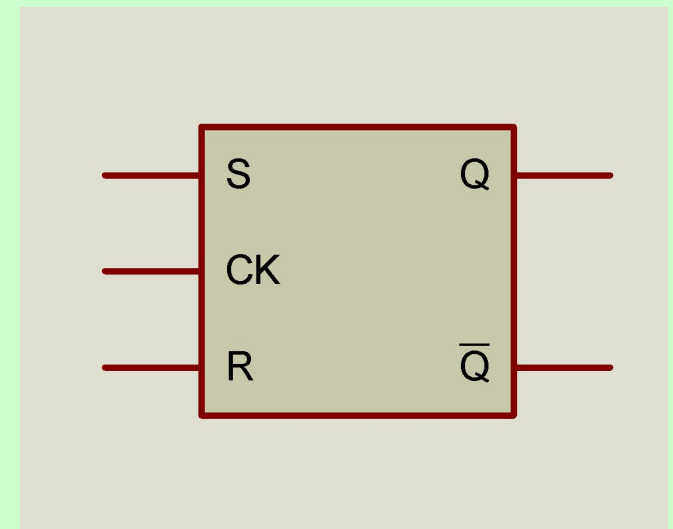
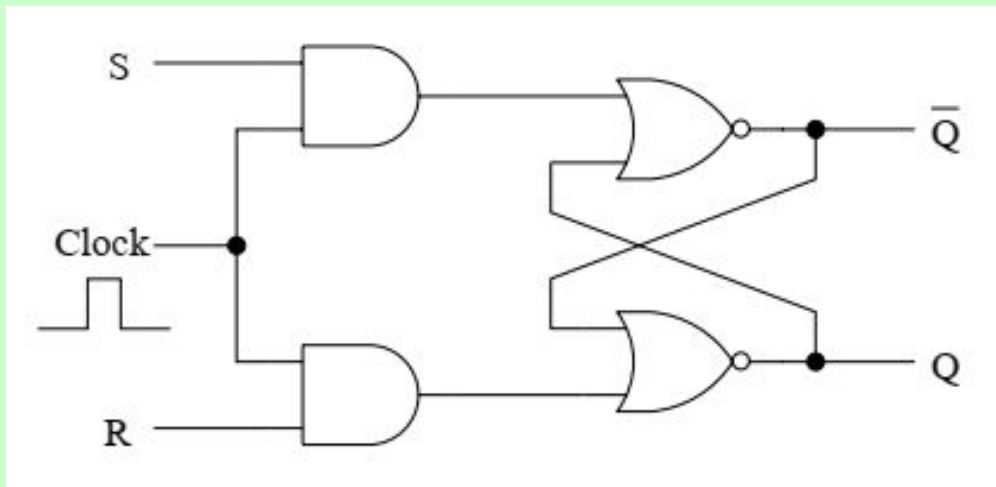
- Q_n represents the current state
- Q_{n+1} represents the next state

S-R Latch As a 1-bit Memory

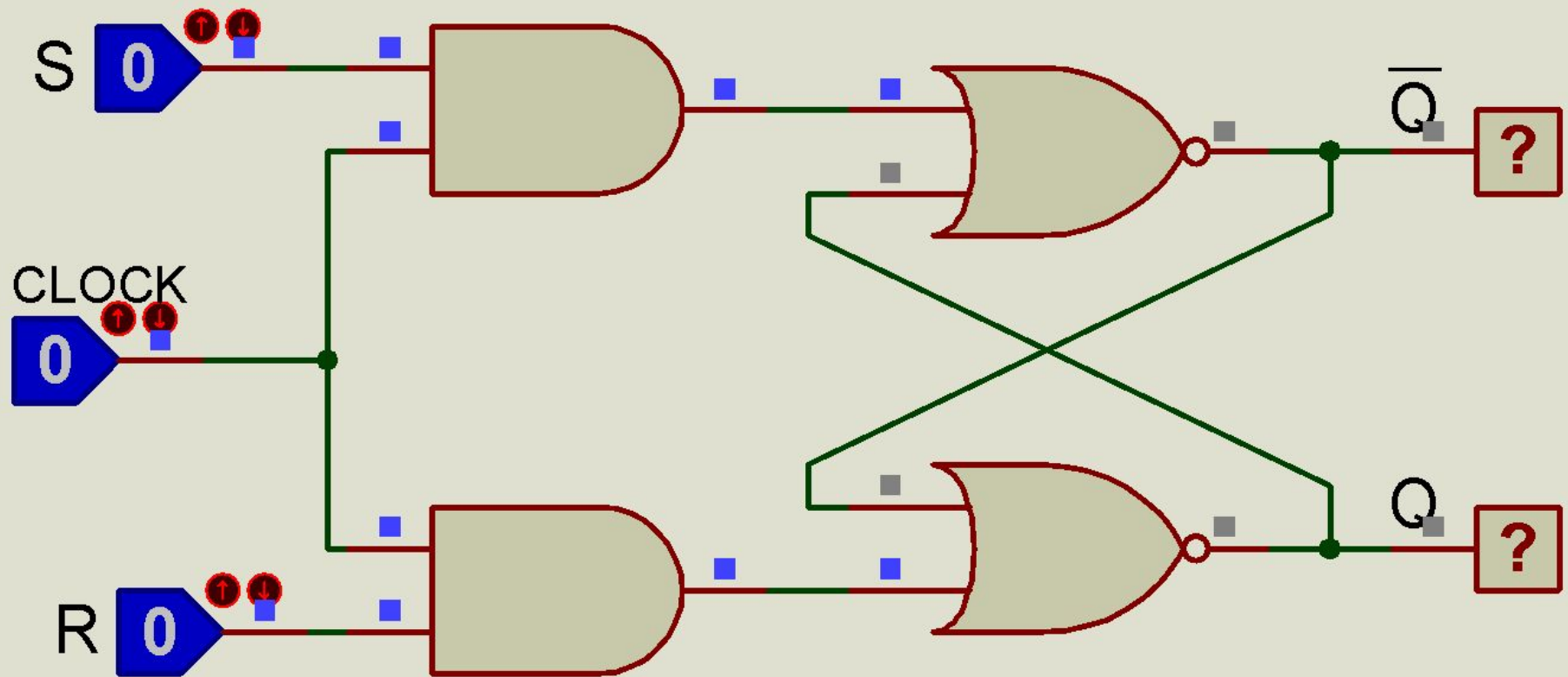
- Latch has the capability to store a bit
- To write 1 into this latch, set SR as 10
- To write 0, use SR = 01
- To retain a stored bit, keep both S and R inputs at 0
- Avoid setting SR to 11
- Once we have the design to store a single bit, we can replicate this circuit to store more bits or bytes.

Clocked S-R Latch

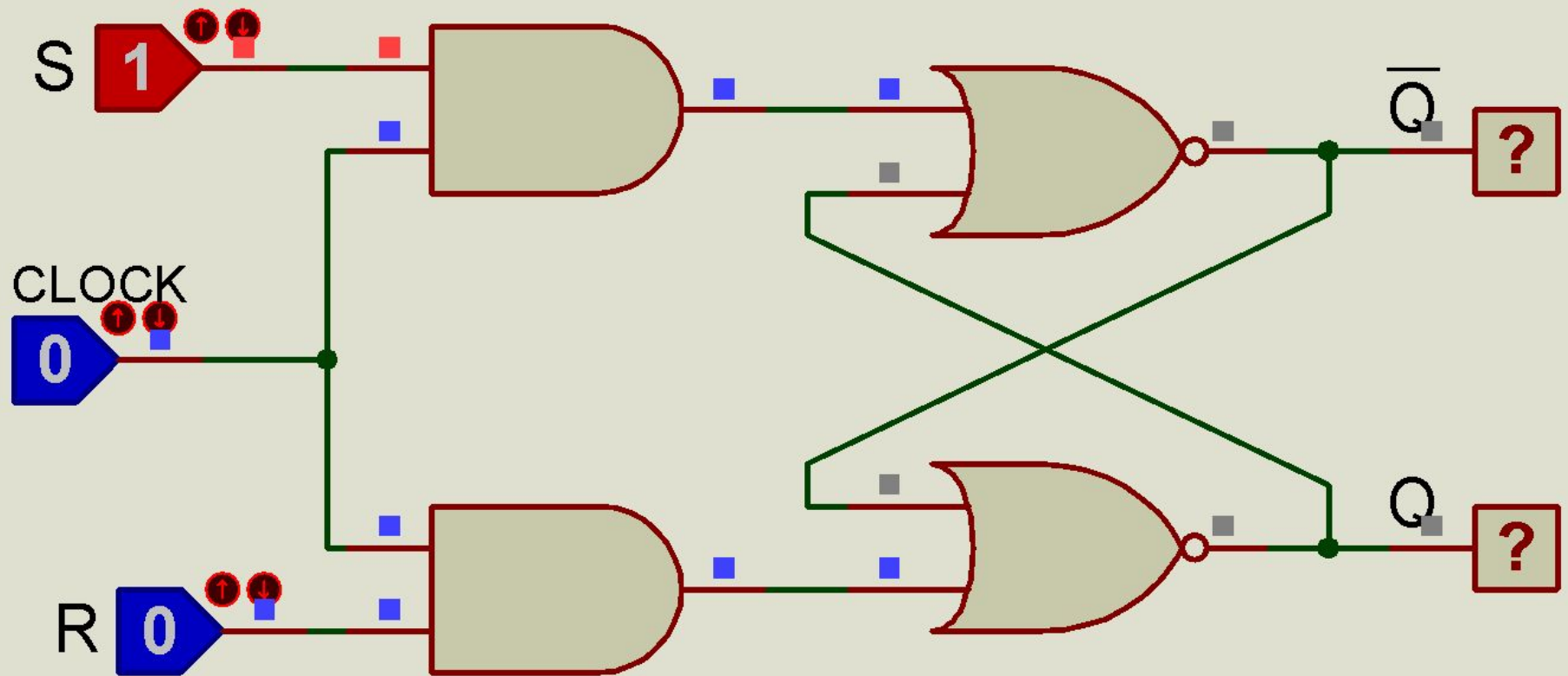
- How to make S-R latch synchronized
 - Output changes at specific instants
- Adding clock to gate the inputs to provide synchronization
- Now it is called clocked S-R Latch



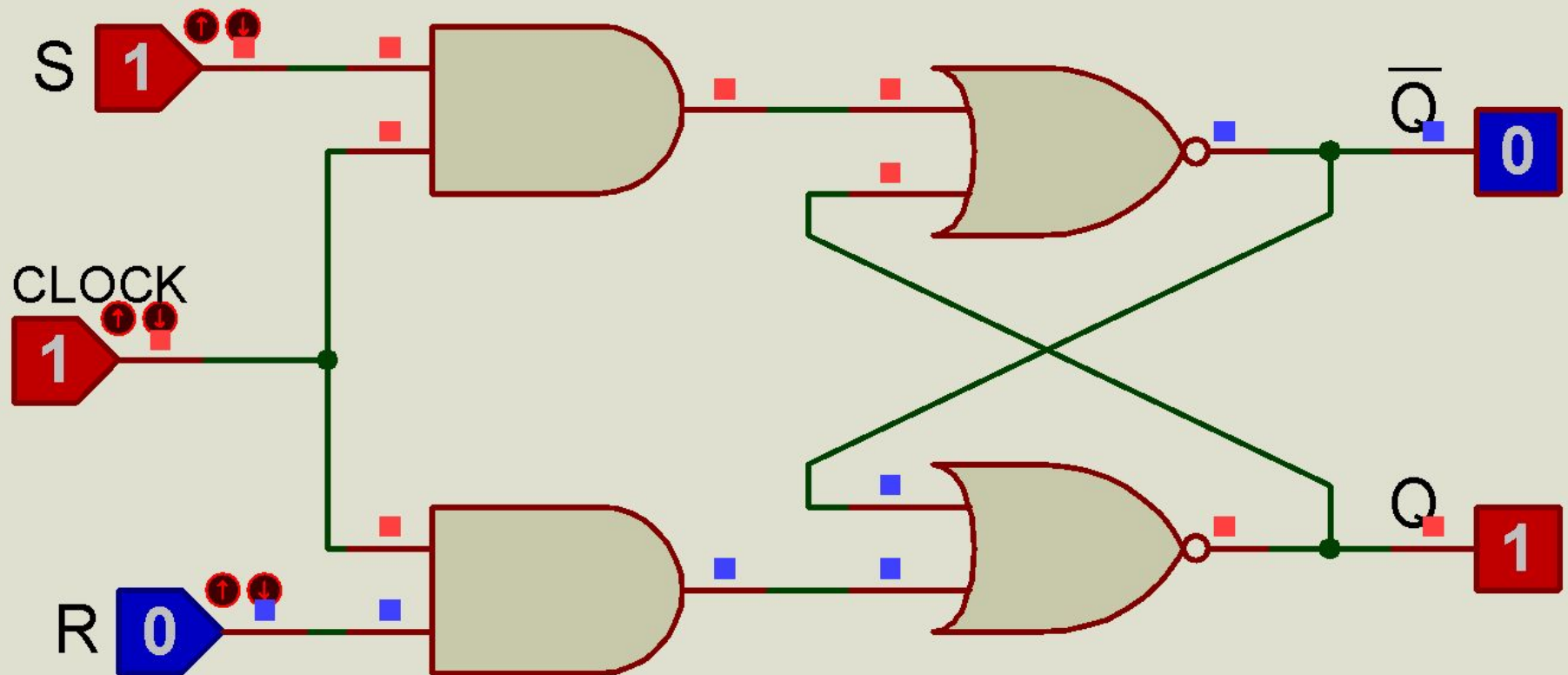
Clocked S-R Latch



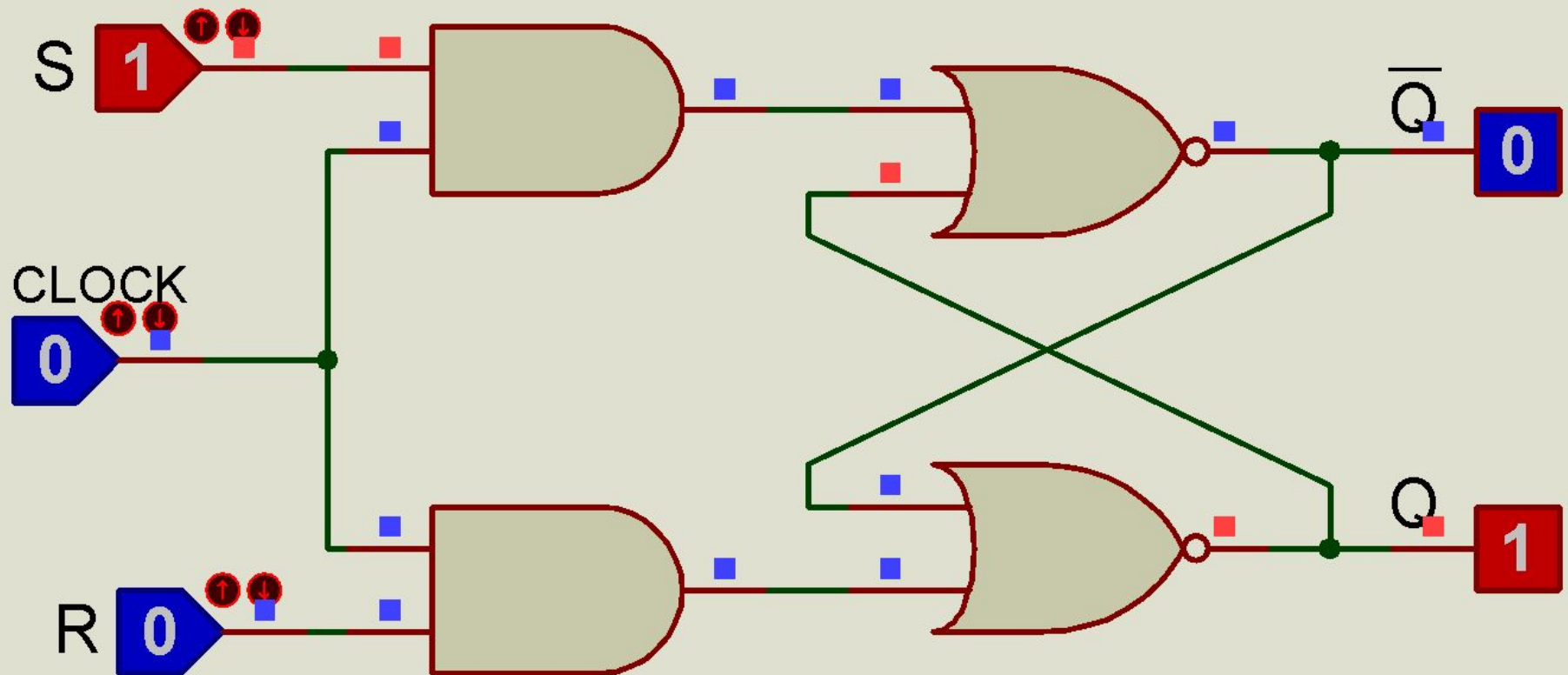
Clocked S-R Latch



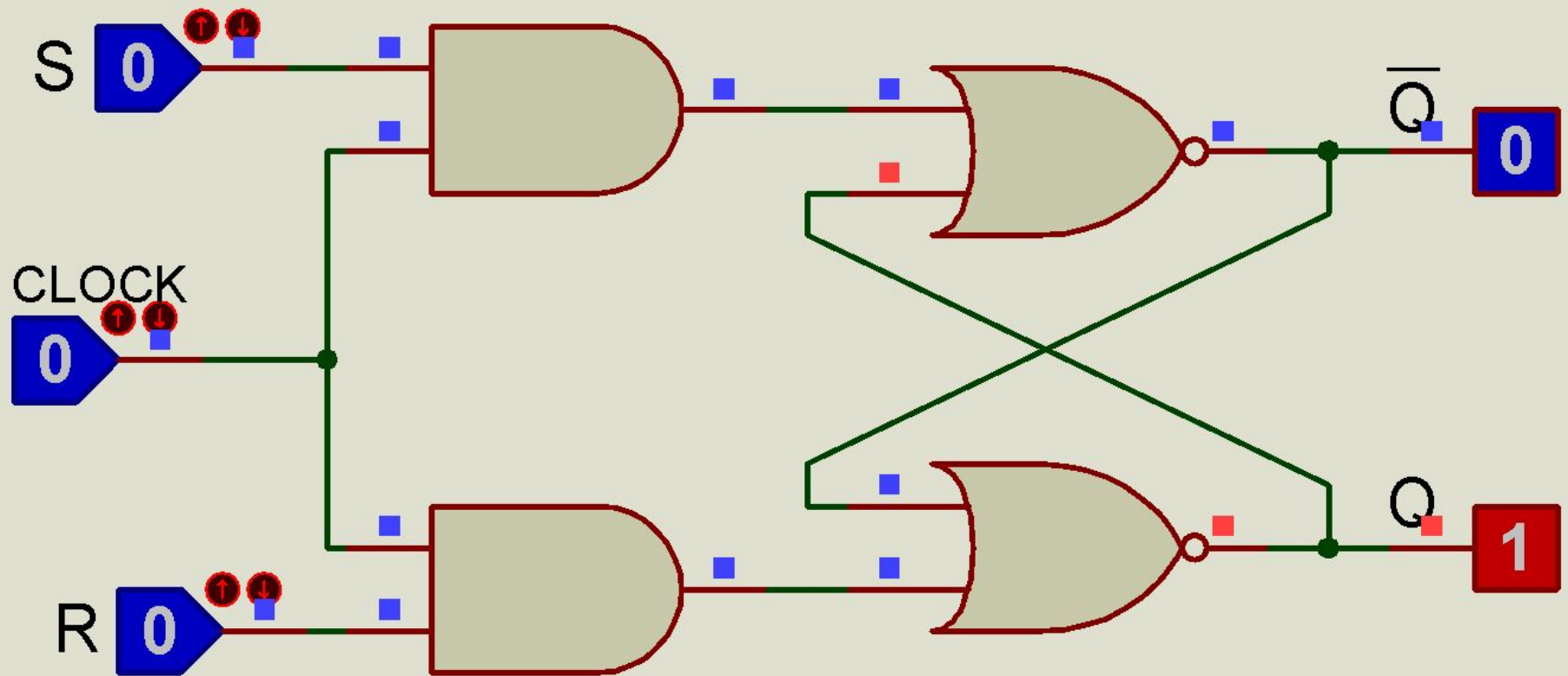
Clocked S-R Latch



Clocked S-R Latch

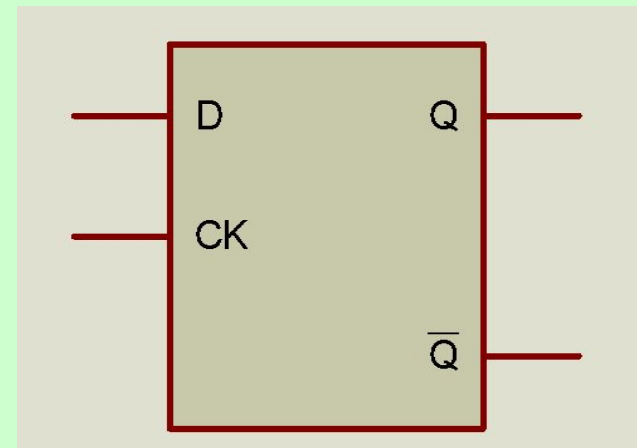
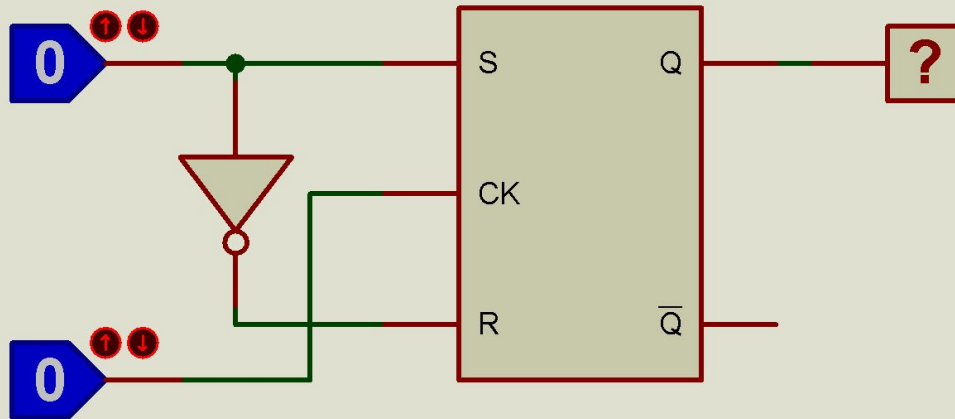


Clocked S-R Latch

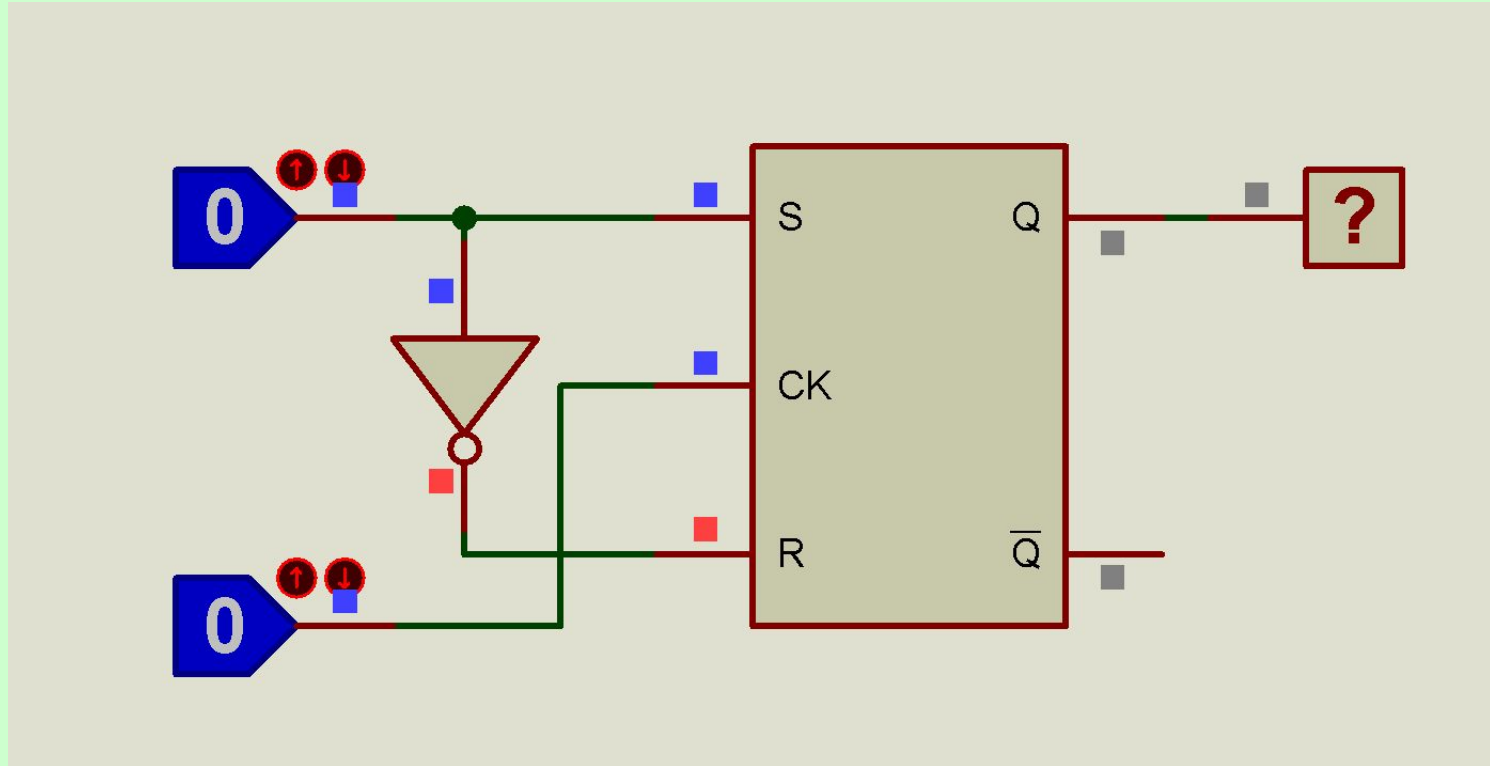


D - Latch

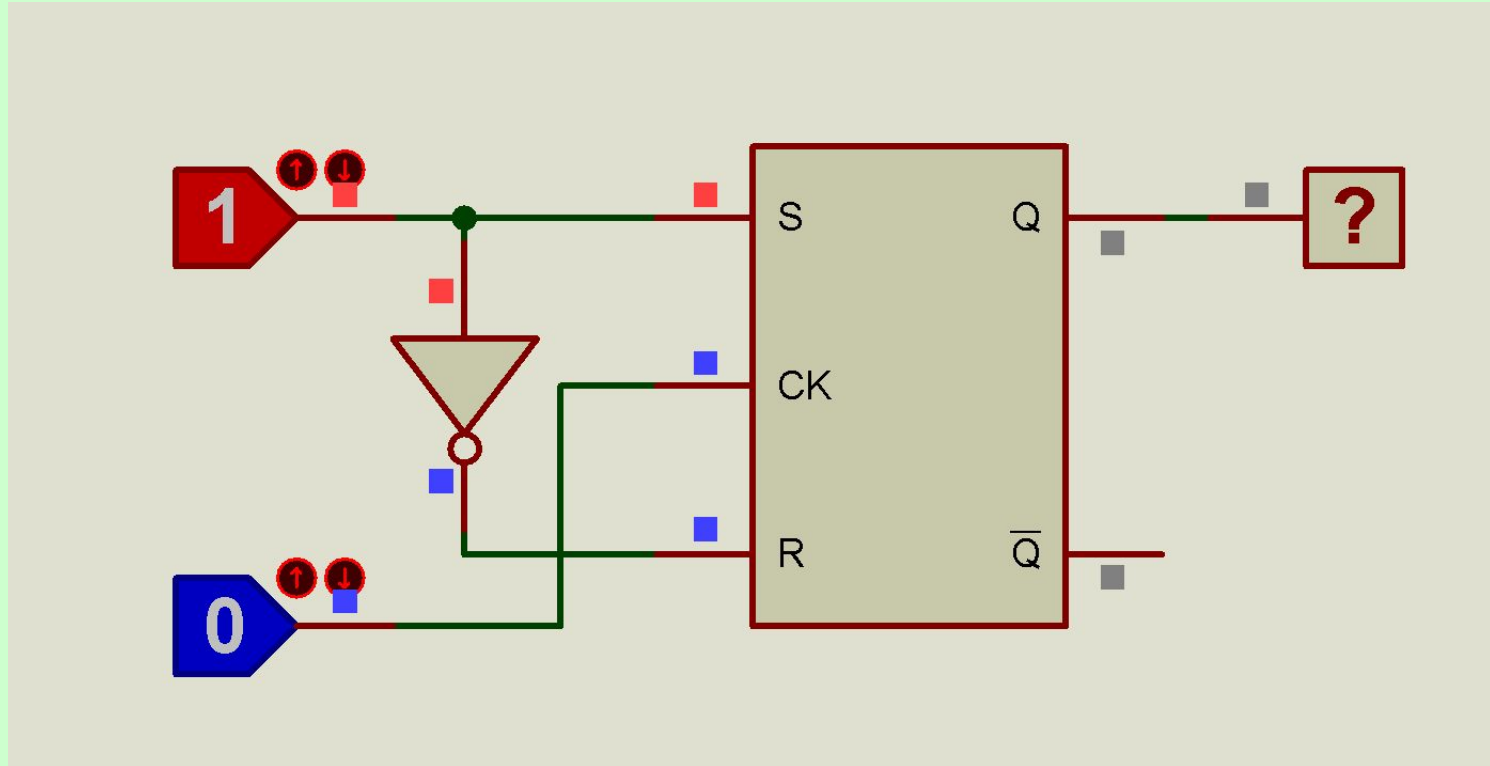
- A problem with SR latches is that we have to avoid the $SR = 11$ input combination
- This problem is solved by the D latch



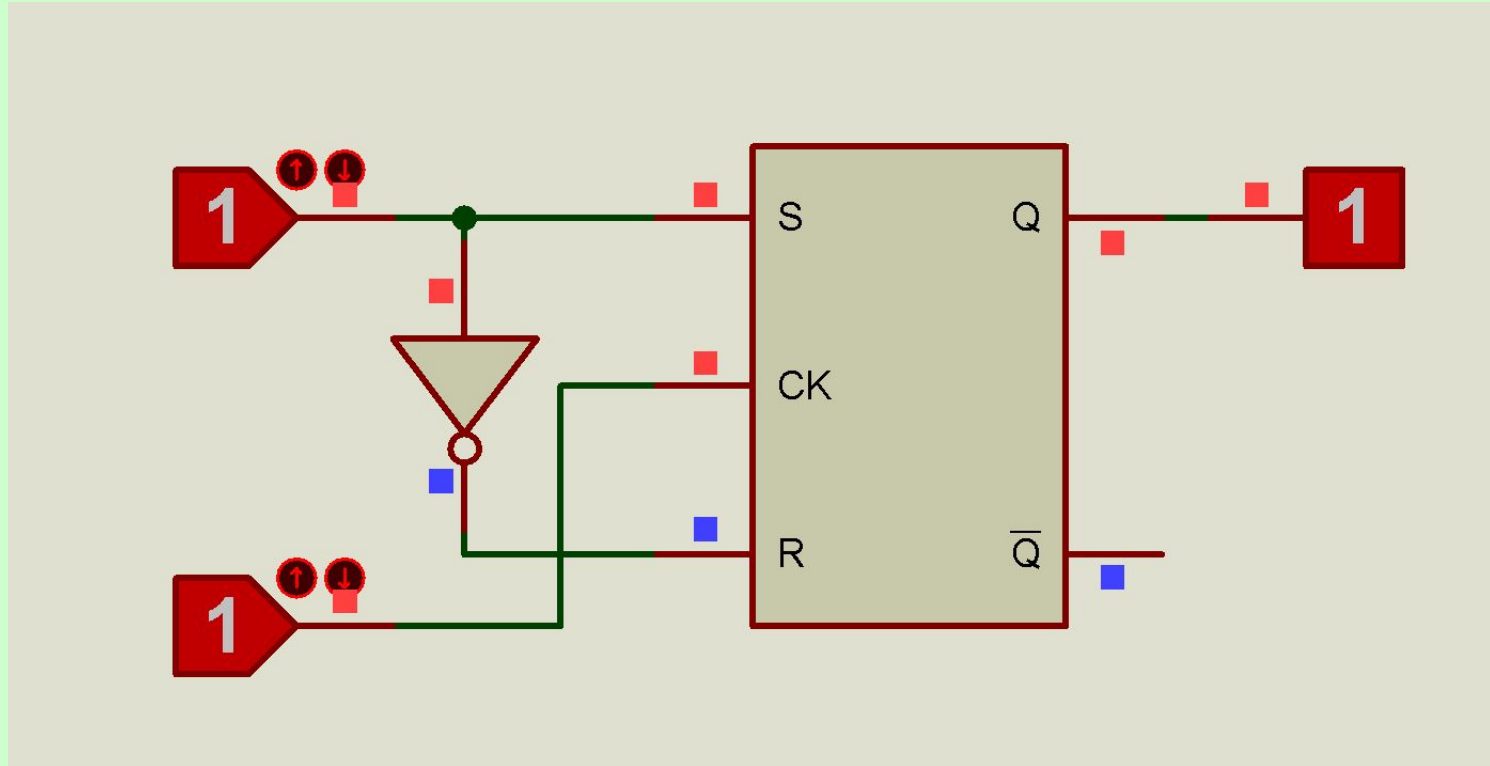
D – Latch Operation



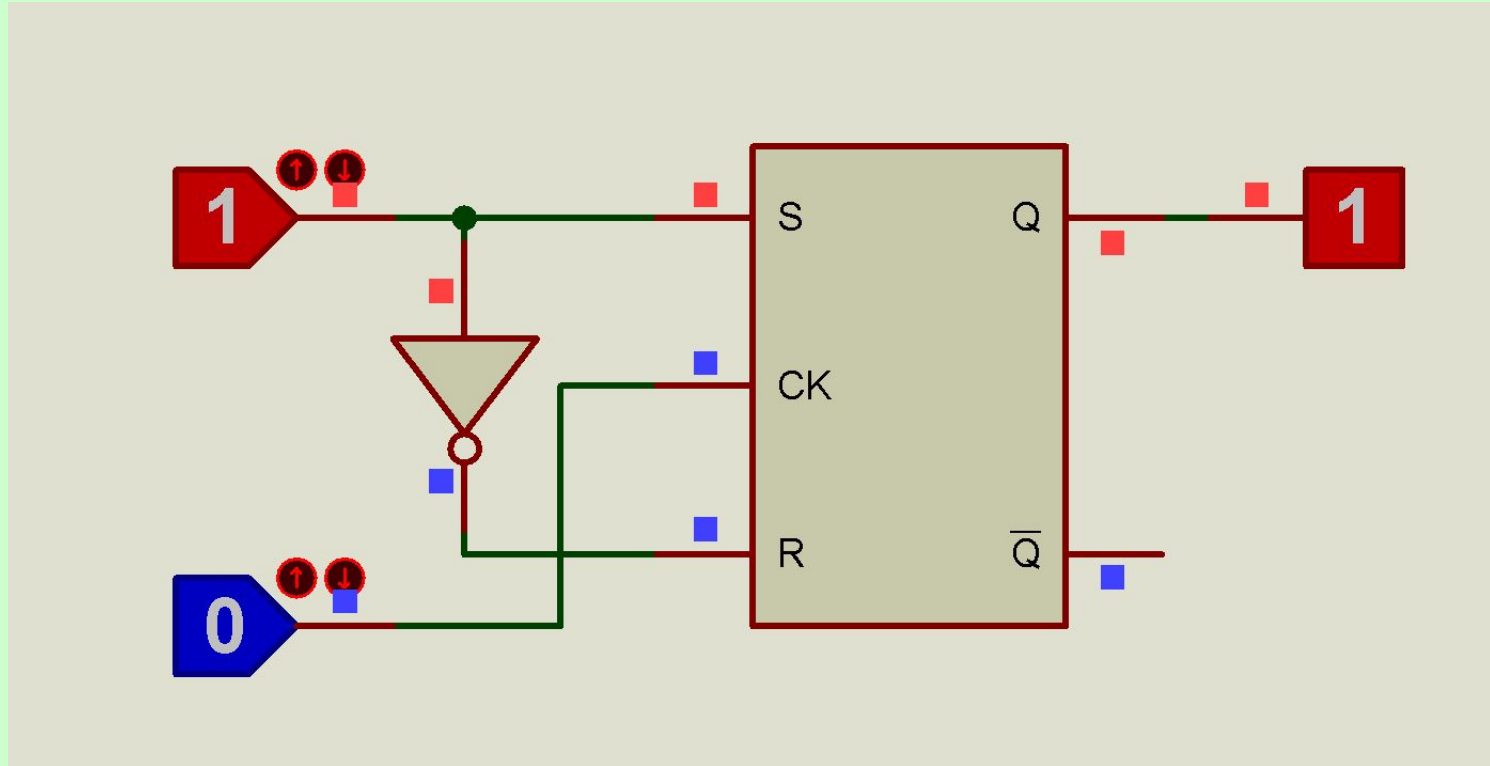
D – Latch Operation



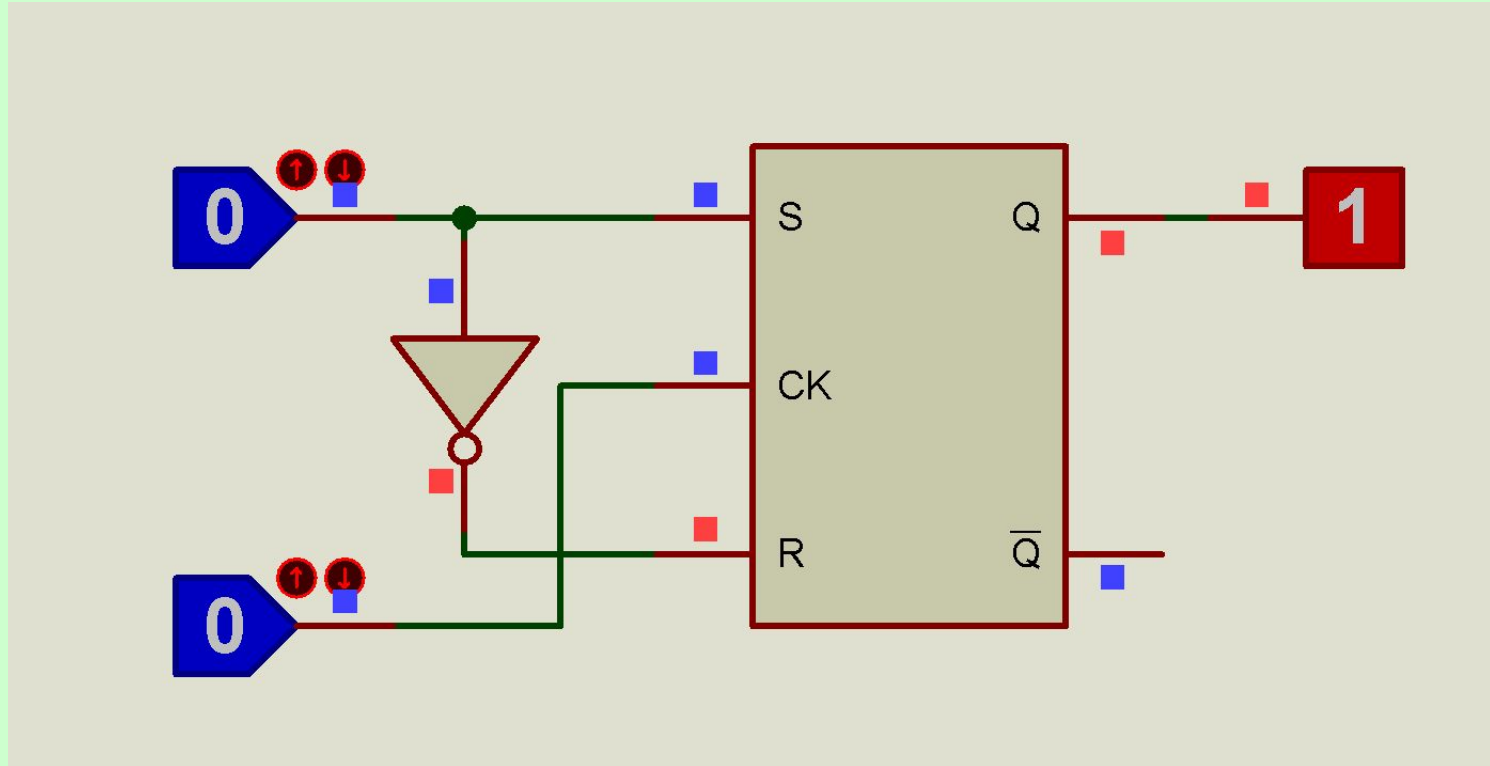
D – Latch Operation



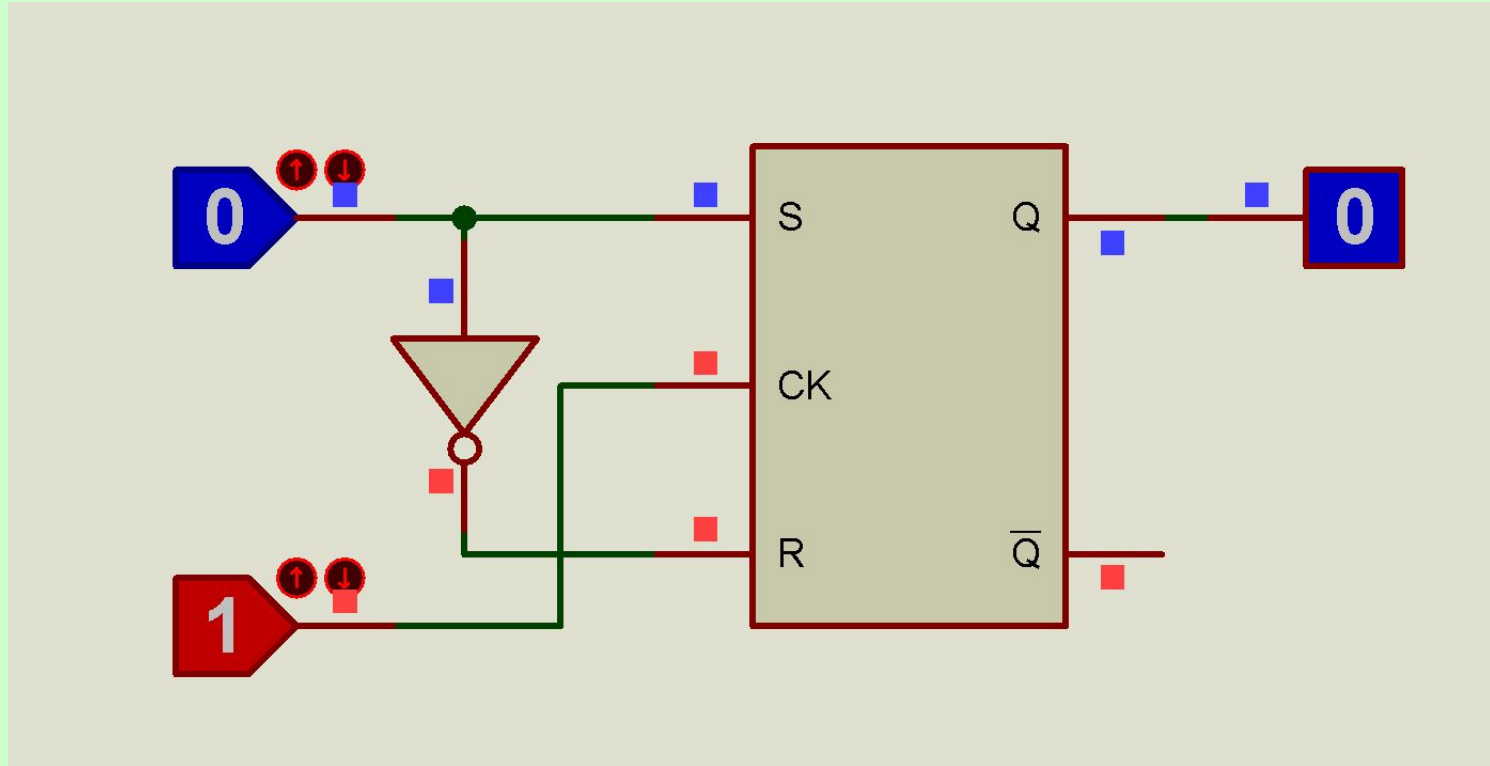
D – Latch Operation



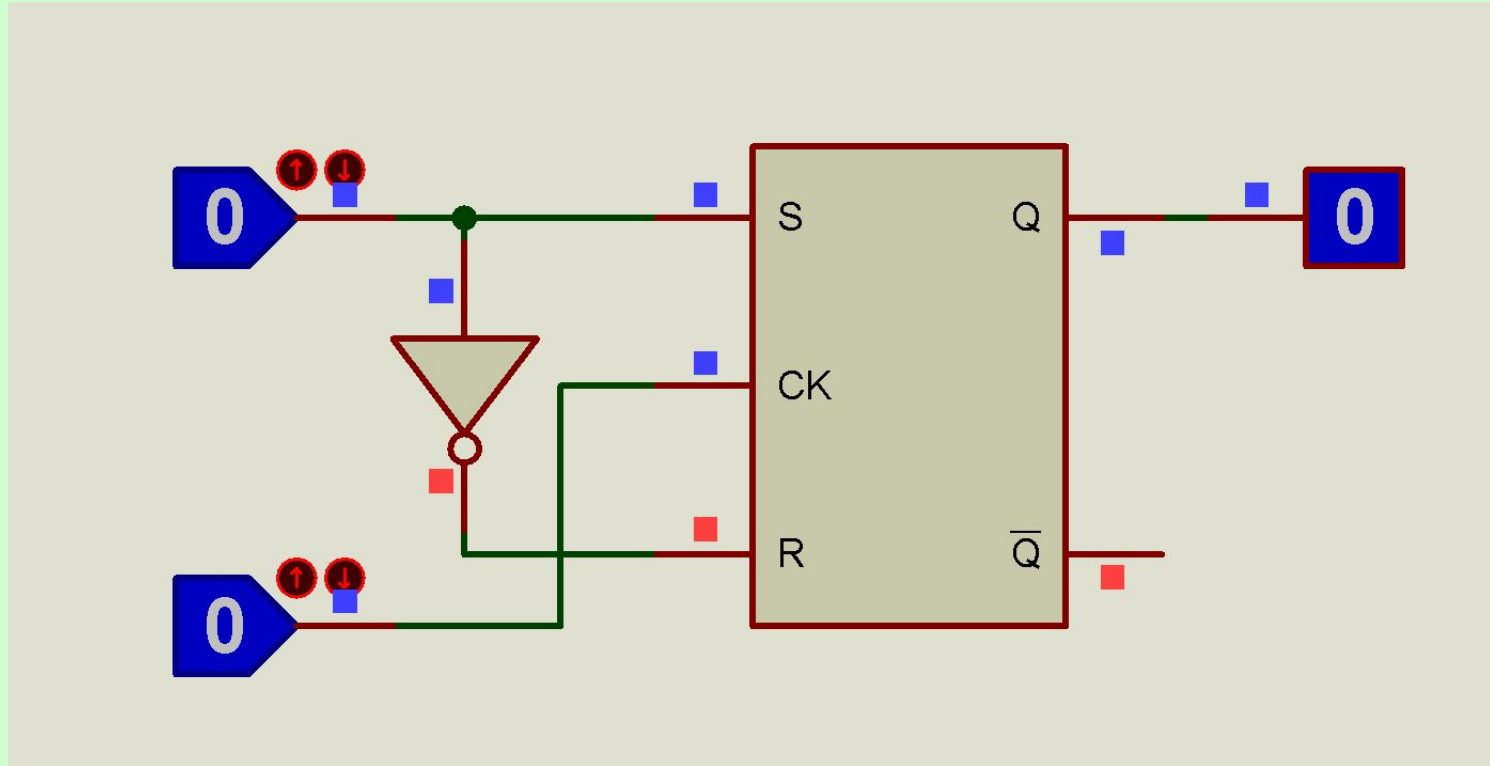
D – Latch Operation



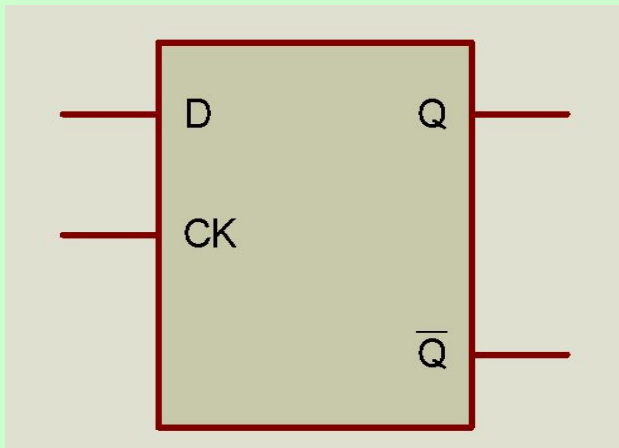
D – Latch Operation



D – Latch Operation



D – Latch Truth Table

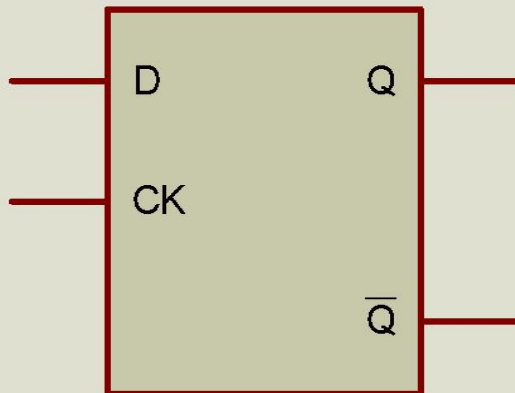


D	Q_{n+1}
0	0
1	1

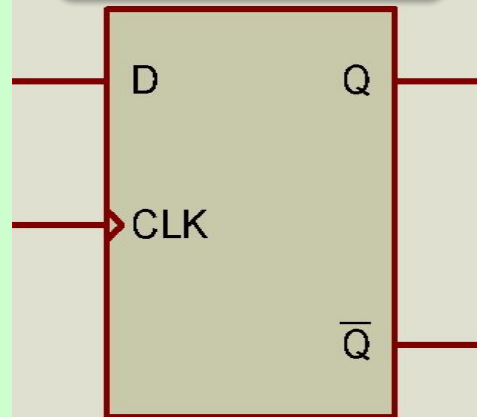
FLIP FLOPS

- Flip-flops are edge-triggered devices whereas latches are level sensitive
- Some manufacturers do not follow this distinction
- How edge triggering can be achieved ?
- One solution is to use Master-Slave Technique
- Note the difference between D-latch and D-FF symbols

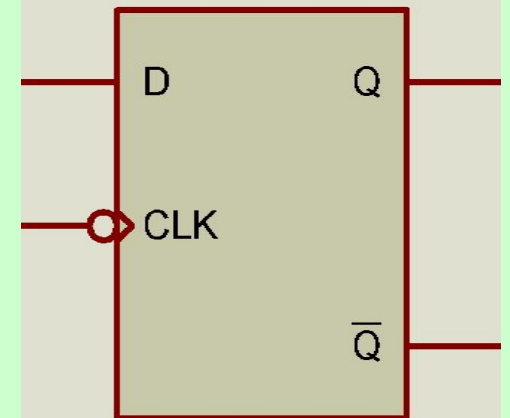
Latch



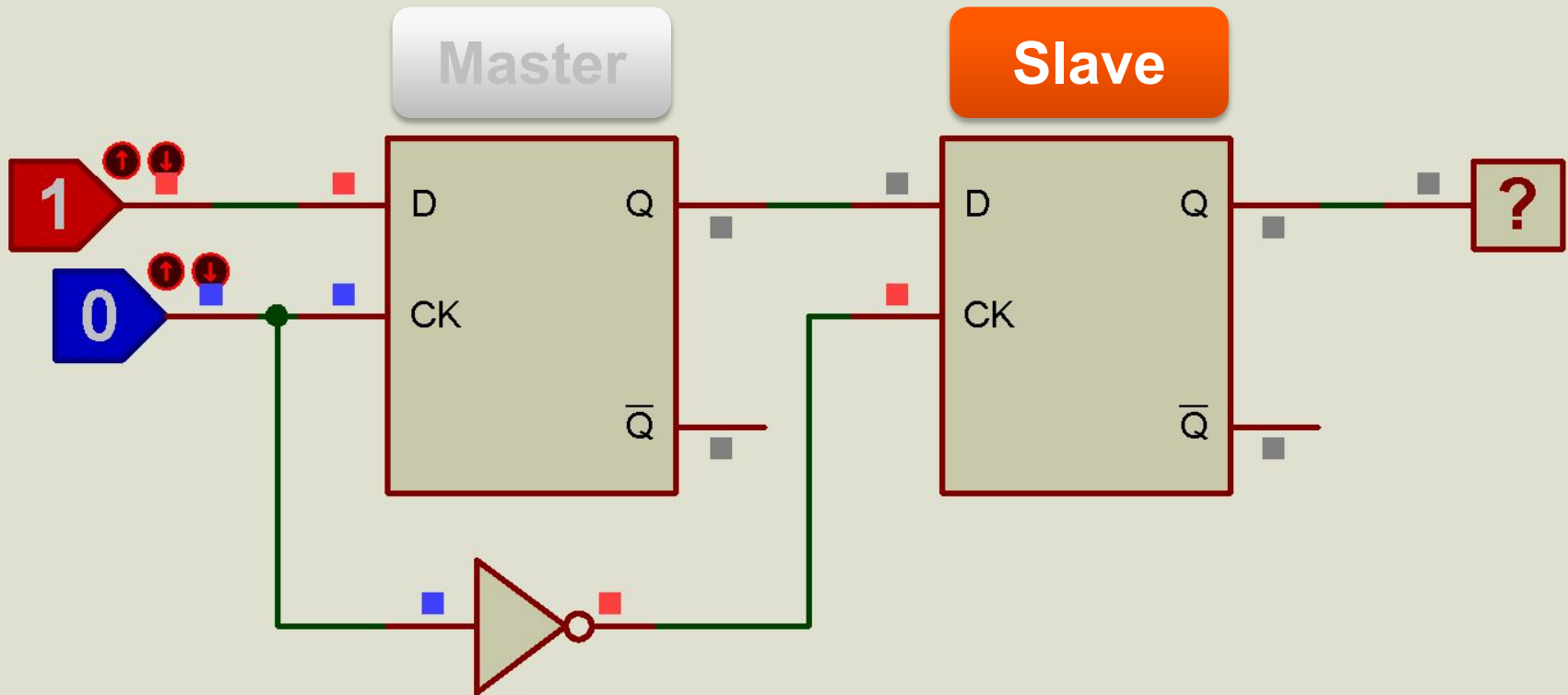
FF +edge



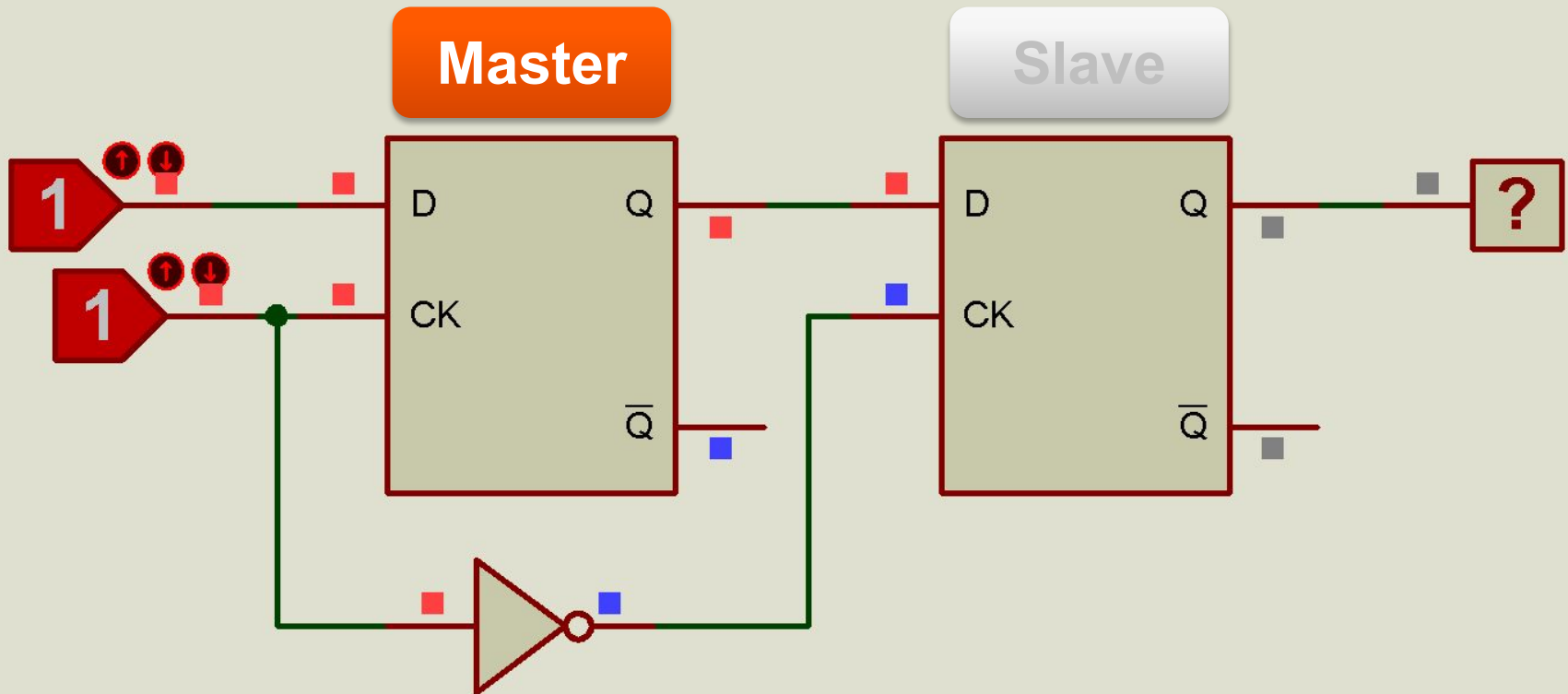
FF -edge



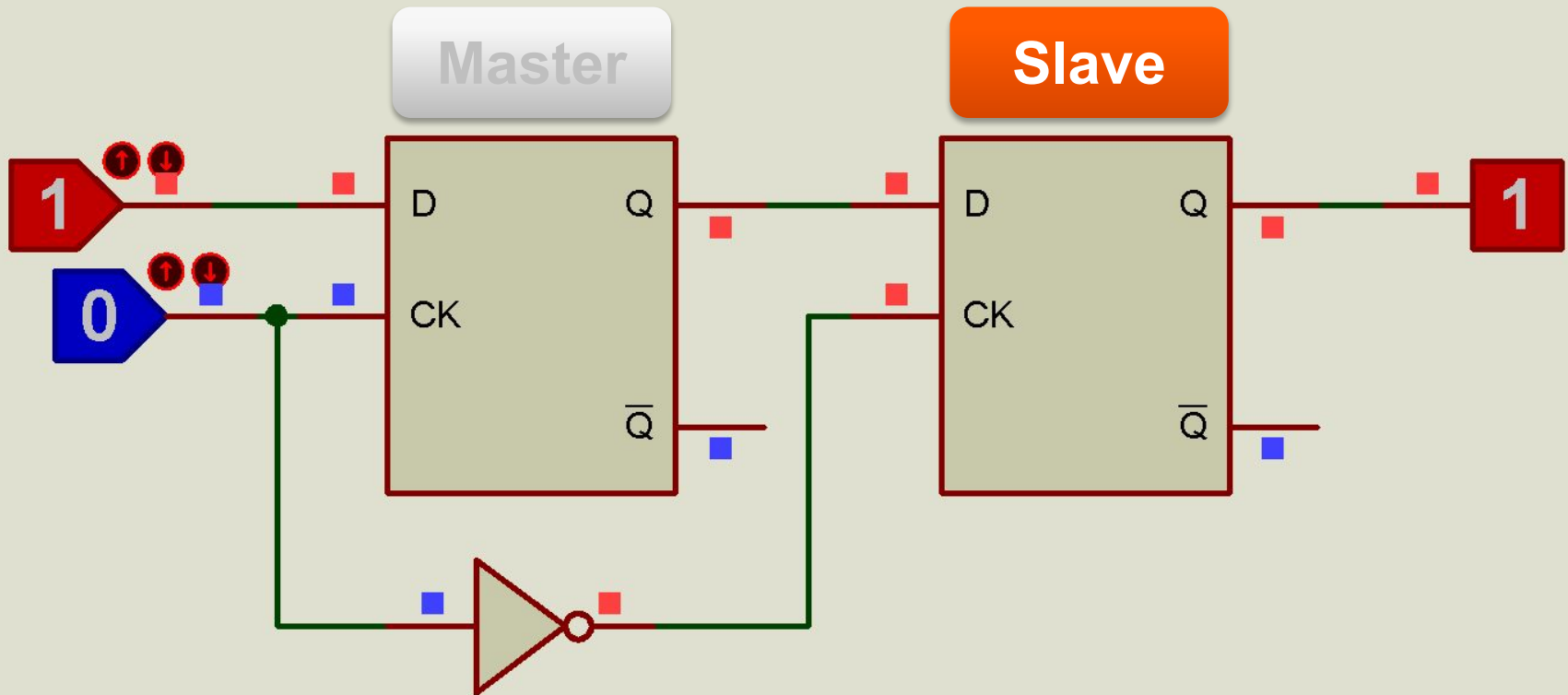
Master – Slave FF



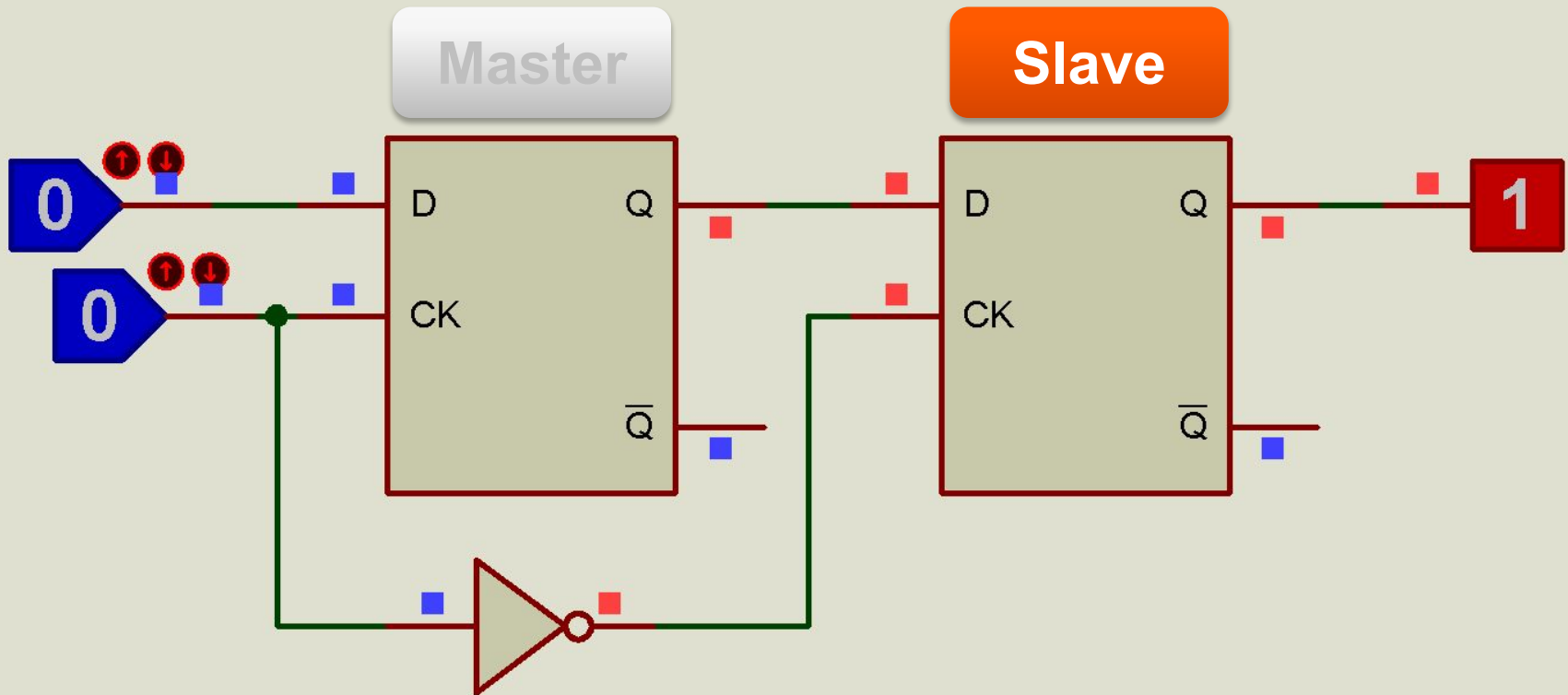
Master – Slave FF



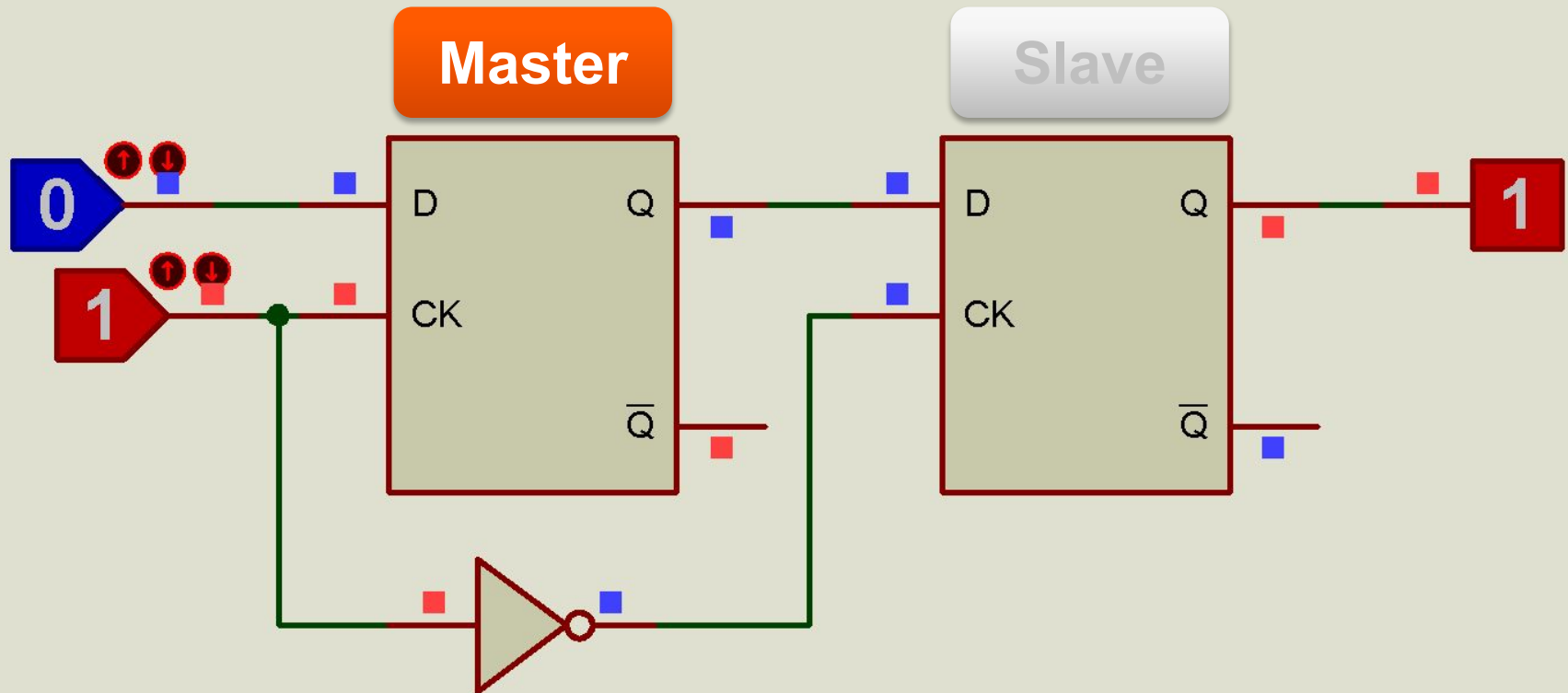
Master – Slave FF



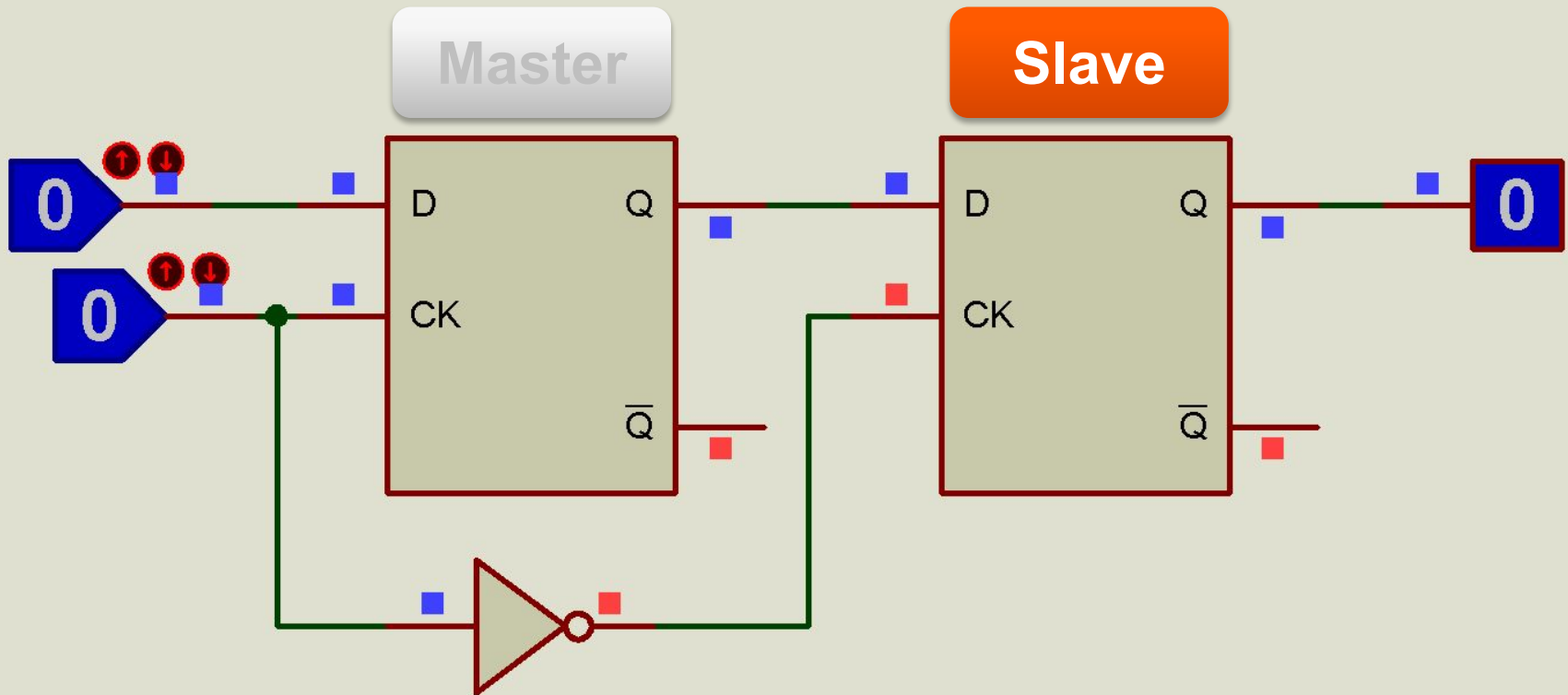
Master – Slave FF



Master – Slave FF

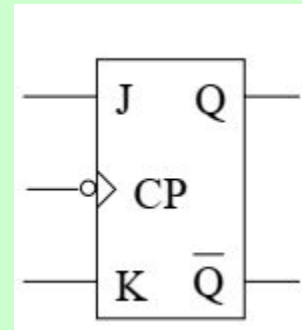
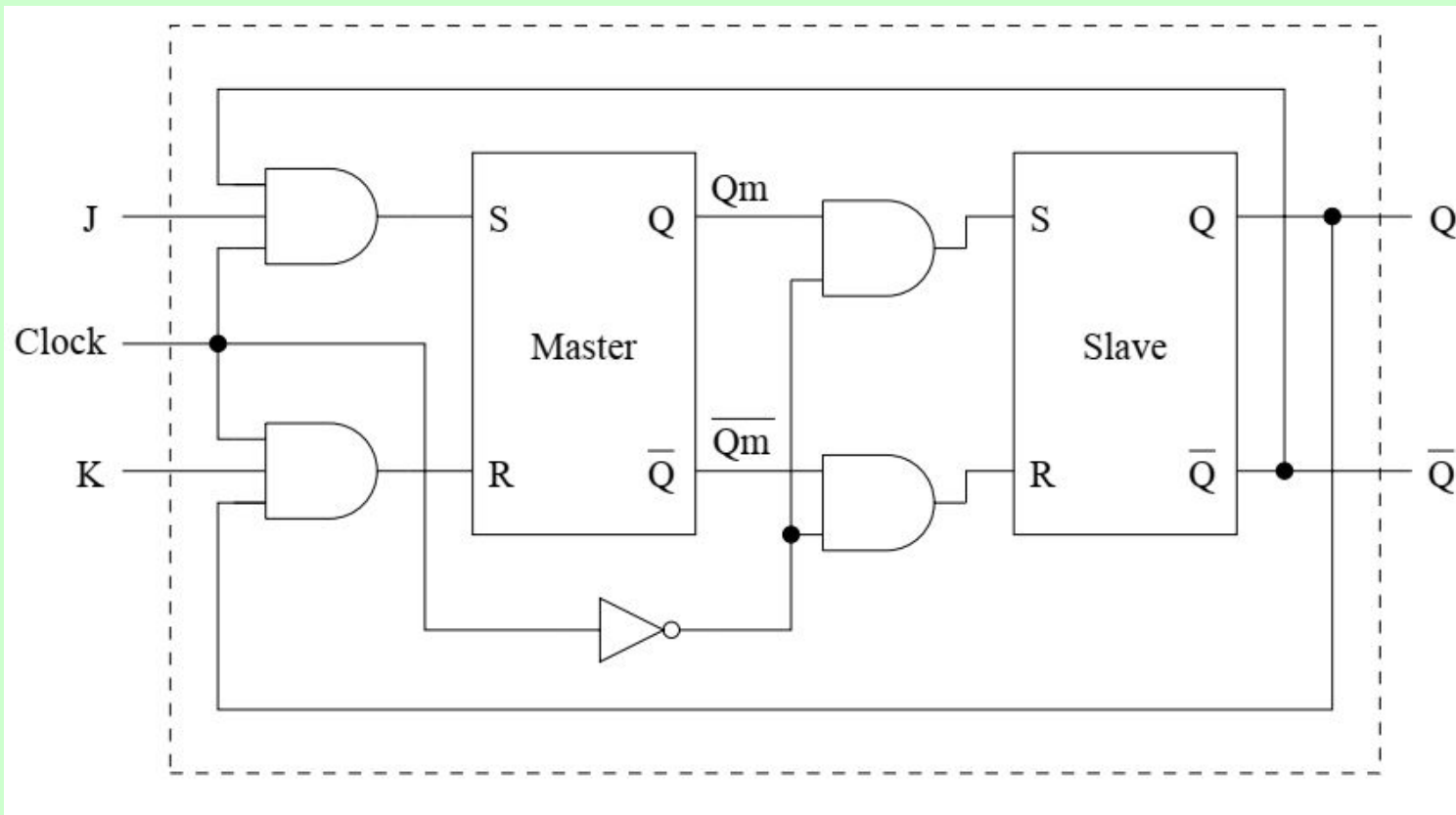


Master – Slave FF



J-K FLIP FLOP

- It is another type of Flip-Flops that makes the avoided input combination “SR”=11 useful
- J-K Flip-Flop uses a 2nd feedback as shown



J-K FLIP FLOP Truth Table

- The truth table of J-K FF indicates that
 - J-K works as S-R FF for the first 3 states (00-01-10)
 - Now the state of JK=1 1 makes $Q_{n+1} = \overline{Q_n}$ (Toggle)

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$

No
Change

Reset

Set

Toggle

J-K FLIP FLOP Example

